

Defence Science Journal, Vol. 56, No. 4, October 2006, pp. 451-463  
© 2006, DESIDOC

## Estimation of Launch and Impact Points of a Flight Trajectory using U-D Kalman Filter/Smoother

V.P.S. Naidu, Girija G., and J.R. Raol

*National Aerospace Laboratories, Bangalore-560 017*

### ABSTRACT

The launch and impact points of a flight trajectory are estimated using U-D Kalman filter and Rauch-Tung-Striebel (R-T-S) smoother. Algorithms are implemented in PC MATLAB and validated using simulated data. The filter performance is evaluated in terms of state error, innovation sequence, and autocorrelation of residuals along with their theoretical bounds. The R-T-S smoother was found to generate accurate state estimates, which led to better launch point estimation. Launch and impact point prediction from real data of a guided target in ballistic mode is also evaluated.

**Keywords:** Trajectory estimation, U-D Kalman filter, R-T-S smoother, Kalman tracking flight trajectory, launch point, impact point, target detection, air defence, Kalman tracking filter

### 1. INTRODUCTION

Air defence applications like weapon delivery systems require an accurate prediction of launch and impact points of targets detected during a part of their trajectory. The launch point information is required to know from where the target was launched and whether the target was from enemy territory or friendly territory. The impact point information is required to know where the target would impact, in the ocean or in a major city, so that appropriate action to destroy the target could be initiated. Kalman tracking filters have been used to predict such trajectories. The advantage of Kalman filter is that along with the position estimates, it gives estimates of the accuracy of the predictions. If the sensors observe the entire flight trajectory, then the estimation of the launch and impact points is straightforward and a Kalman filter could be used for the estimation. However, there could be situations where the flight

trajectory is available only for a part of its flight. The estimation of impact and launch points in such a situation would require an extrapolation of the estimated trajectory backward in time to predict the launch point and forward in time to predict the impact point.

Estimation of the launch and impact points have been carried out in two phases. In the first phase where the moving object is observed by sensors, its trajectory is estimated using point-mass model and a Kalman filter<sup>1</sup>. Smoothing techniques significantly improve the initial condition of the state estimates<sup>2</sup>, this could be used to predict the launch point accurately, and hence, smoothed state estimates are obtained using Rauch-Tung-Striebel (R-T-S) smoother during the first phase. In the second phase where the target is unobservable by sensors, the trajectory is estimated using point-mass model and covariance propagation calculation by

forward propagation to get information about the impact point of the target<sup>3</sup>. The smoothed state estimate at the start time of the data is used to predict the launch point of the trajectory using backward integration.

In this study, U-D factorised form of the Kalman filter has been chosen as it has the advantage of being numerically stable and computationally efficient. U-D factorisation-based Kalman filter and forward prediction were used to predict the impact point. U-D factorisation-based Kalman filter and fixed-interval Rauch-Tung-Streibal (R-T-S) smoother and backward integration were used to predict the launch point. The probable use of U-D filter is in its application to real-time tracking<sup>4</sup>. Algorithms were implemented in PC MATLAB and validated using simulated data of a target moving with constant velocity. The algorithms were also used to predict launch and impact points from real data of a flight trajectory for which these points were known by considering certain segments of the data as being observable by the sensors. Results have been presented in terms of the accuracy of prediction of the launch and impact points, time history comparisons, autocorrelation of the residuals with bounds, innovation sequence with the theoretical bounds<sup>5</sup>, and state error with bounds for the simulated data.

## 2. U-D FACTORISATION-BASED KALMAN FILTER AND RTS SMOOTHER

The two phases involved in the estimation of the launch and impact points have been illustrated in Fig. 1. A typical trajectory of a target from A to D is shown with A as the launch point and D as the impact point. Assuming that measurements are available only between points B to C where sensors are tracking the target, it is proposed to estimate the impact point D and launch point A. In the first phase, data between B and C was used to generate estimated states and covariances using U-D filter. An R-T-S smoother working between points B and C utilising the output of the Kalman filter-generated smoothed states in a backward pass (*ie*, going from C to B) operation. In the second phase, point D was obtained using the forward prediction of the U-D filter output at point C, and point A was estimated by backward integration starting with the smoother output at point B.

### 2.1 U-D Kalman Filter

The general form of the kinematic state model of the tracked target is given by

$$x(k+1) = \Phi x(k) + Gw(k) \quad (1)$$

with discrete measurements

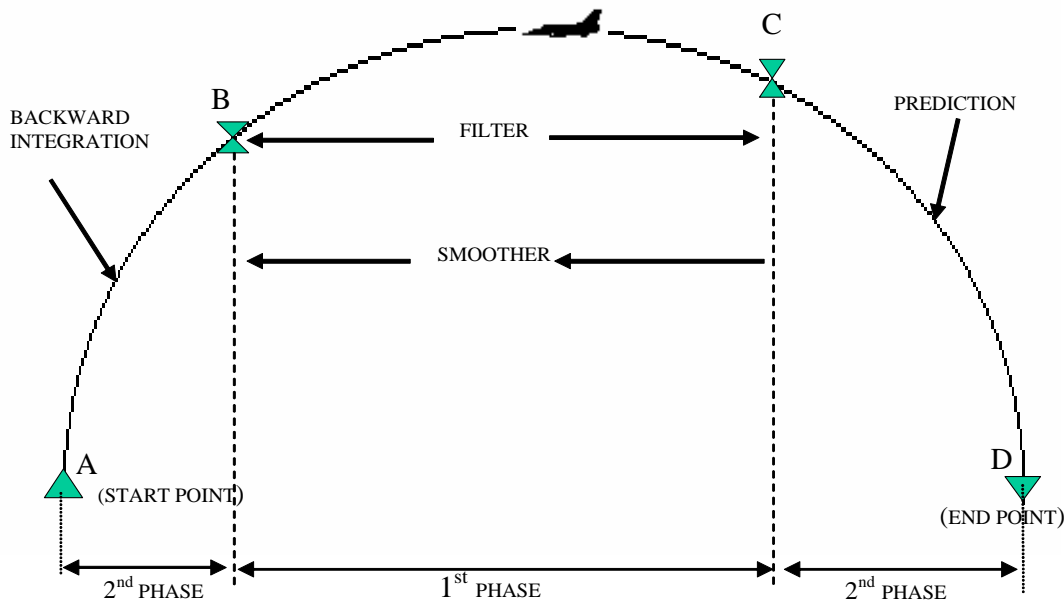


Figure 1. Typical target trajectory.

$$\begin{aligned} z(k+1) &= H x(k+1) + v(k+1) \\ k &= 0, 1, 2, \dots, N \end{aligned} \quad (2)$$

where,  $x$  is the state vector,  $z$  is the measurement vector,  $\Phi$  is the state transition matrix,  $G$  is the process noise related matrix,  $H$  is the observation matrix,  $v$  is the measurement noise,  $w$  is the process noise, and  $N$  is the number of measurements.

It was assumed that the noise sequences  $\{w(k)$  and  $v(k)\}$  are zero mean white Gaussian and are mutually independent and have diagonal covariances  $Q(=\sigma_w^2)$  and  $R(=\sigma_v^2)$  respectively. The choice of  $G$  assures that  $Q$  being diagonal is not a loss of generality and by Cholesky decomposition whitening, it can be assumed with no loss of generality that  $R$  is diagonal<sup>4</sup>.

The term U-D covariance factorisation comes from a property of non-negative definite symmetric matrices, according to which covariance matrix  $P$  can be factored into  $P = UDU^T$ , where  $U$  is the upper triangular matrix with unit elements on its main diagonal and  $D$  is a diagonal matrix<sup>4</sup>. Covariance and gain processing algorithms, operating on  $U$  and  $D$  factors of state error covariance matrix  $P$ , are a technique for implementing square root filtering without requiring computation of square roots. The U-D Kalman filtering algorithm is considered efficient, stable, and accurate for real-time applications<sup>4</sup>. The U-D factorisation-based Kalman filtering algorithm is given in two parts namely time propagation and measurement update algorithms.

## 2.2 U-D Factor Time Propagation Algorithm

State estimate extrapolation:

$$\tilde{x}_{k+1/k} = \Phi \hat{x}_{k/k} \quad (3)$$

Error covariance extrapolation:

$$\tilde{P}_{k+1/k} = \Phi \hat{P}_{k/k} \Phi^T + GQG^T \quad (4)$$

Given  $\hat{P} = \hat{U}\hat{D}\hat{U}^T$  and  $Q$  as the process noise covariance matrix, the time update factors  $\tilde{U}$  and  $\tilde{D}$  are obtained through modified weighted Gram-Schmidt

orthogonalisation process. Defining  $W = [\Phi\hat{U} \mid G_A]$ ,  $\bar{D} = \text{diag}[\hat{D}, Q]$  with  $W^T = [w_1, w_2, \dots, w_n]$ , where  $n$  is the number of states in state vector.  $P$  is reformulated as  $\tilde{P} = \tilde{W}\tilde{D}\tilde{W}^T$ . The  $U$  and  $D$  factors of  $\tilde{W}\tilde{D}\tilde{W}^T$  may be computed as described below.

For  $j = n, n-1, \dots, 2$ , the following equations are recursively evaluated:

$$\tilde{D}_j = \langle w_j^{(n-j)}, w_j^{(n-j)} \rangle_{\bar{D}}$$

$$\tilde{U}(i, j) = \langle w_i^{(n-j)}, w_j^{(n-j)} \rangle_{\bar{D}} / \tilde{D}_j$$

$$w_i^{(n-j+1)} = w_i^{(n-j)} - \tilde{U}(i, j)w_j^{(n-j)}, i = 1, \dots, (j-1)$$

$$\tilde{D}_1 = \langle w_1^{(n-1)}, w_1^{(n-1)} \rangle_{\bar{D}} \quad (5)$$

## 2.3 U-D Factor Measurement Update Algorithm

The measurement update in Kalman filtering combines a priori estimate  $\tilde{x}$  and  $\tilde{P}$  error covariance with scalar observation  $z = a^T x + v$ ;  $a^T = H$  to construct an updated (filtered state) estimate and covariance:

$$K = \tilde{P}_{k+1/k} a / \alpha$$

$$\hat{x}_{k+1/k+1} = \tilde{x}_{k+1/k} + K(z(k+1) - a^T \tilde{x}_{k+1/k})$$

$$\alpha = a^T \tilde{P}_{k+1/k} a + r$$

$$\hat{P}_{k+1/k+1} = \tilde{P}_{k+1/k} - Ka\tilde{P}_{k+1/k} \quad (6)$$

where  $k = N_B, \dots, N_C$ ,  $\tilde{P} = \tilde{U}\tilde{D}\tilde{U}^T$ ,  $a$  is the measurement vector,  $r$  is the measurement noise covariance,  $z$  is the string of noisy measurements,  $N_B$  and  $N_C$  denote start and end of the data measurements as available from the tracking sensors.

Kalman gain  $K$  and updated covariance factors  $\hat{U}$  and  $\hat{D}$  can be obtained from the following equations:

$$f = \tilde{U}^T a; \quad f^T = (f_1, \dots, f_n)$$

$$v = \tilde{D}f; \quad v_i = \tilde{d}_i f_i \quad i = 1, 2, \dots, n$$

$$\hat{d}_1 = \tilde{d}_1 r / \alpha_1; \quad \alpha_1 = r + v_1 f_1;$$

$$K_2^T = (v_1 \ 0 \dots 0) \quad (7)$$

For  $j = 2, \dots, n$  recursively, the following equations are evaluated:

$$\alpha_j = \alpha_{j-1} + v_j f_j; \quad \hat{d}_j = \tilde{d}_j \alpha_{j-1} / \alpha_j$$

$$\hat{u}_j = \tilde{u}_j + \lambda_j k_j; \quad \lambda_j = -f_j / \alpha_{j-1}$$

$$K_{j+1} = K_j + v_j \tilde{u}_j \quad (8)$$

where  $\tilde{U} = [\tilde{u}_1, \dots, \tilde{u}_n]$ ,  $\hat{U} = [\hat{u}_1, \dots, \hat{u}_n]$  and Kalman gain is given by  $K = K_{n+1} / \alpha_n$  where  $\tilde{d}$  is predicted diagonal element and  $\hat{d}_j$  is the updated diagonal element of the  $D$  matrix.

## 2.4 R-T-S Smoother

Smoothing is a non-real time data processing scheme that uses all measurements starting from  $T$  to 0 to estimate the state of a system at a certain time  $t$ , where  $0 \leq t \leq T$ . The smoothing algorithm implemented here is a backward-pass sequel to a forward pass Kalman filter outputs (filter covariances, Kalman gains, etc). The R-T-S smoother formulation is considered, because of the simplicity<sup>2,4</sup>. The R-T-S recursions that generate smoothed estimates and error covariances are:

$$x_{k/N}^* = \hat{x}_{k/k} + G_k^* (x_{k+1/N}^* - \hat{x}_{k+1/k}) \quad (9)$$

$$P_{k/N}^* = \hat{P}_{k/k} + G_k^* (P_{k+1/N}^* - \tilde{P}_{k+1/k}) G_k^{*T} \quad (10)$$

where

$$G_k^* = \hat{P}_{k/k} \Phi_k^T \tilde{P}_{k+1/k}^{-1} \quad (11)$$

The recursion is backward sweep from  $k = (N_C - 1), \dots, N_B$ . The state estimates covariances  $\tilde{x}_{k+1/k}$  and  $\tilde{P}_{k+1/k}$  are found using the U-D filter in the forward pass.

## 2.5 Impact-point Prediction

Forward prediction is used to predict the impact point D (whose index is  $N$ )

$$\tilde{x}_{k+1/k} = \Phi \hat{x}_{k/k} \quad (12)$$

$$\tilde{P}_{k+1/k} = \Phi \hat{P}_{k/k} \Phi^T, \quad k = N_C, \dots, N \quad (13)$$

The initial value of  $\hat{x}_{k/k}$  is the output of the U-D filter at point C.

## 2.6 Launch Point Prediction

Backward integration was carried out to predict the launch point.

$$x_{k/N}^* = \Phi x_{k+1/N}^* \quad (14)$$

$$P_{k/N}^* = \Phi^{-1} P_{k+1/N}^* (\Phi^T)^{-1}, \quad k = (N_B - 1), \dots, 1 \quad (15)$$

The initial value of  $x_{k+1/k}^*$  is the smoother output at point B or U-D filter output at point B.

## 2.7 Filter-performance Evaluation

The filter performance is ascertained by checking (i) the estimated states and the bounds for convergence, (ii) residuals and their bounds for convergence, and (iii) the autocorrelation of the residuals for whiteness. The performance of U-D Kalman filter has been evaluated by checking whether

- The state error  $(x - \hat{x})$  falls within the theoretical bound of  $\pm 2\sqrt{\hat{P}_{(k,k)}}$ , where  $\hat{P} = \hat{U} \hat{D} \hat{U}^T$  (for simulated data only)
- The innovation  $\varepsilon = z - H\tilde{x}$  falls within the theoretical bound of  $\pm 2\sqrt{S_{(k,k)}}$ , where  $S = H\tilde{P}H^T + R$

- The autocorrelation of residuals fall within the theoretical bound of  $\pm 1.96/\sqrt{N}$ , where  $N$  is the number of samples.

### 3. RESULTS & DISCUSSION

The U-D factorised Kalman filter and R-T-S smoother techniques are implemented in PC MATLAB. The performance of the algorithm has been evaluated using simulated data of a target moving with constant velocity. It is then applied to real data for extrapolating the estimated/smoothed trajectory during a segment of the flight trajectory.

#### 3.1 Validation Using Simulated Data

For generating the simulated data, the following state variables of the target are considered:  $x$ -position,  $x$ -velocity,  $y$ -position, and  $y$ -velocity, *ie.* the state vector is represented by  $[x, \dot{x}, y, \dot{y}]$ . Equations (1) and (2) give the model of the system considered. Sampling time  $\Delta T = 0.1$  s is used to simulate data for 50 s., resulting in a total of 500 data points. The transition matrix and the other related matrices are given by:

$$\Phi = \begin{pmatrix} 1 & \Delta T & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad G = \begin{pmatrix} \Delta T^2 / 2 & 0 \\ 0 & \Delta T \\ \Delta T^2 / 2 & 0 \\ 0 & \Delta T \end{pmatrix}$$

$$R = \begin{pmatrix} r_x^2 & 0 \\ 0 & r_y^2 \end{pmatrix} \quad H = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad Q = \begin{pmatrix} q_x^2 & 0 \\ 0 & q_y^2 \end{pmatrix} \quad (16)$$

The initial state vector  $x_0 = [1200, 8.3, 3000, 10.6]$ . Measurements of  $x$ -position and  $y$ -position are generated using Eqn (2) by adding random noise with variance of  $r_x = 1$  and  $r_y = 100$ , respectively. The process noise variance used in the simulation is  $q_x = q_y = 0.01$ . The results of estimation are the average of 10 Monte Carlo simulations. Figure 2 shows the  $x$  and  $y$  position state estimates using the data between the points B and C (in the left-half of the figure) assuming that measured data is available between 10 s and 40 s. The state estimates with prediction of launch point (of point A) using the smoother output as well as the filter output and

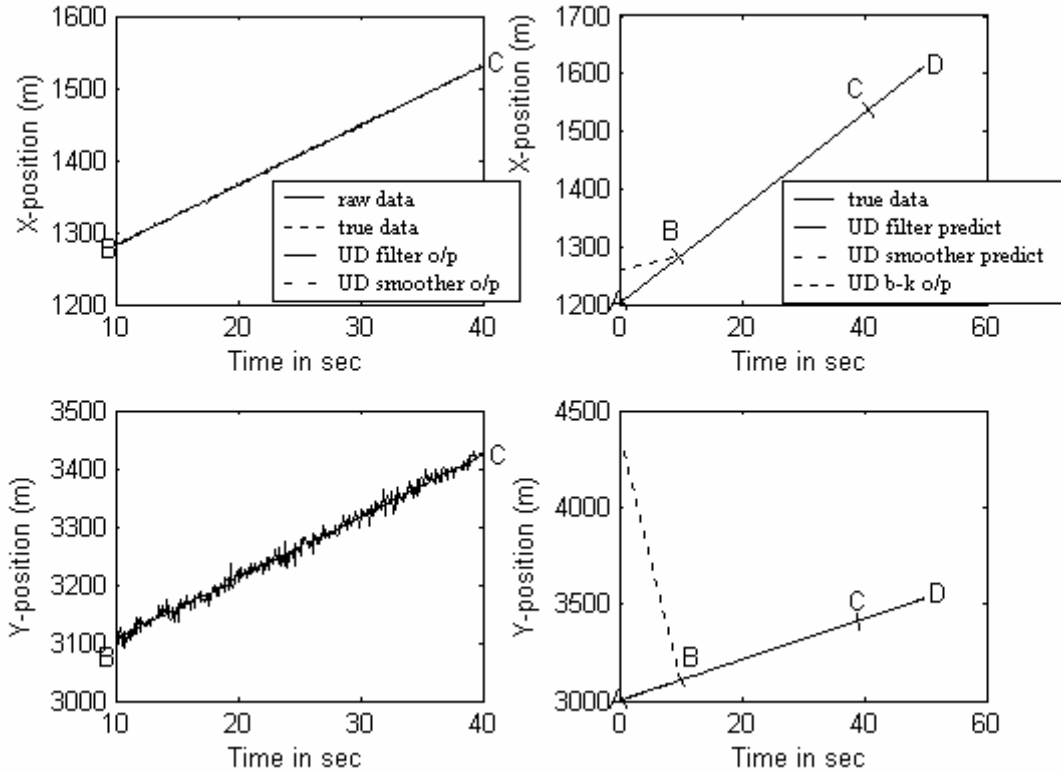


Figure 2. Estimated states of both filter and smoother for simulated data.

impact point (point D) using the filter output are compared with the true states in the right-half of the Fig. 2.

It is clear from the figure that the launch-point prediction using U-D Kalman filter is away from the true value. However, the smoother gives an accurate launch-point prediction. The innovations state errors and autocorrelation of residuals with bounds are shown in Fig. 3. The innovations, residuals, and autocorrelation function fall within the theoretical

bounds, which indicate satisfactory filter performance. The results of impact-point prediction using the U-D Kalman filter output at point C is shown in Table 1. A number of cases are presented where it is assumed that measured data is available for different time segments (indicated by points B and C in the Tables).

Results are presented in terms of the impact point prediction error, the standard deviation of the impact-point prediction, the mean of the residuals and the fit error. It is clear that even with only 10

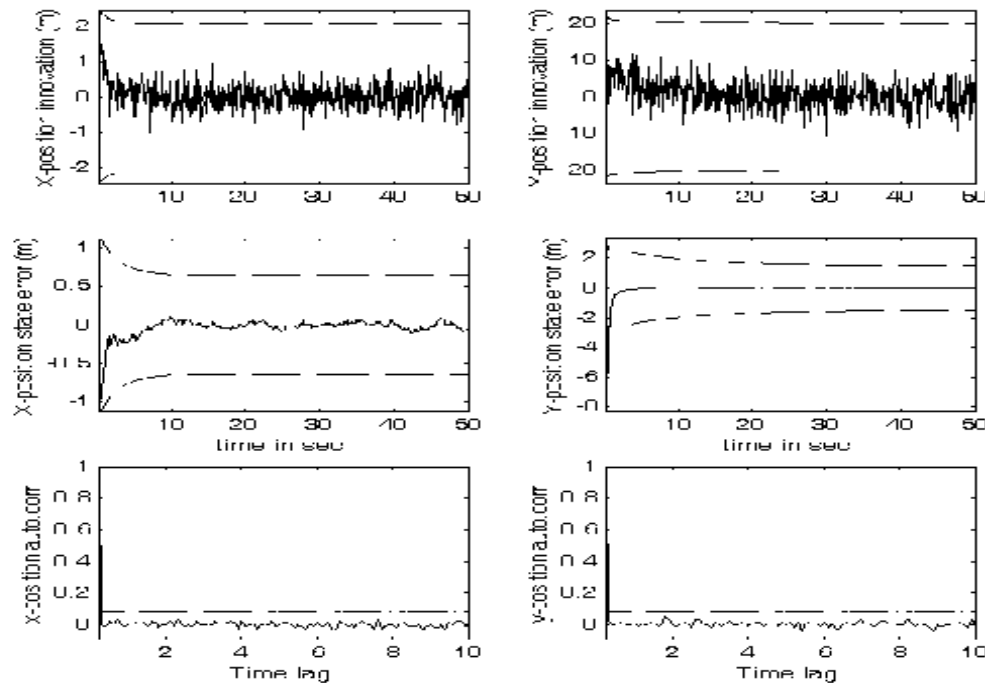


Figure 3. Innovation, state error and autocorrelation with respective bounds.

Table 1. Impact point prediction using UD filter – simulated data  
(POINT-D (x-TRUE=1613.77, Y-TRUE=3528.77), AT 50 s)

Case	Time at B (s)	Time at C (s)	x-pre	x-std	% x err	y-pre	y-std	% y err	mx	my	PFE x	PFE y
1	0.0	50	1613.8	0.20	-0.002	3528.6	1.19	0.006	0.0019	-0.13	0.068	0.298
2	0.5	49.5	1613.7	0.23	0.005	3528.4	1.23	0.01	-0.006	-0.375	0.069	0.299
3	2.5	47.5	1613.8	0.34	-6E-04	3528.4	1.41	0.009	0.0307	0.762	0.068	0.299
4	5.0	45	1613.5	0.50	0.019	3529.1	1.67	-0.01	0.0395	0.391	0.068	0.295
5	10	40	1613.5	0.82	0.017	3530.0	2.28	-0.035	0.0652	1.036	0.068	0.291
6	20	30	1613.5	0.82	0.015	3531.6	2.73	-0.08	0.1044	0.454	0.064	0.277

x, y-pre = predicted x, y-position, x, y-std = standard deviation of x, y-position prediction,

x, y-err = x, y-position error =  $((x, y\text{-true}) - (x, y\text{-pre}))$ , % x, yerr =  $(x, y\text{-err} * 100)/(x, y\text{-true})$ ,

mx, my = mean of x, y residuals, PFE x, y = Percentage fit error in x, y-position.

s of data (case 6), the impact-point prediction is accurate, thereby proving that when the model of the target and the noise statistics are known accurately, the prediction of the impact point using U-D Kalman filtered output is accurate.

The results of launch-point prediction using the U-D Kalman filter state estimate at point B and the smoother output at point B are presented in Table 2. It is clear that in all the cases, the use of the R-T-S smoother results in prediction with better accuracy than using the filter alone. Also, it is clear from Table 2 that the launch-point prediction accuracy degrades considerably when smaller segments of data are observed by the sensors. However, use of U-D Kalman filter with smoother is able to overcome this problem in launch-point prediction accuracy, which is clear from Table 3. The results in all cases are presented in bar graph form in Figs 4(a), 4(b) and 4(c) respectively. The simulated data of a target moving with constant velocity was

chosen for validation of the technique/algorithm. The application of this technique to real data could involve nonlinear state equations and manoeuvring targets. This problem would require an extensive modelling effect.

### 3.2 Application to Real Data

The U-D Kalman filter and smoothing algorithms are used to predict the launch and impact points of a guided target trajectory in ballistic mode. The launch point and impact point for this data are known. To evaluate the accuracy of predictions, when only a part of the trajectory is tracked by the sensors, a number of segments of the data are considered. The trajectory data is measured in terms of  $x$ ,  $y$ , and  $z$  positions in Cartesian frame of reference. The state variables of the start vehicle are  $X$  coordinate position ( $x$ ), velocity ( $\dot{x}$ ), and acceleration ( $\ddot{x}$ ),  $Y$  coordinate position ( $y$ ), velocity ( $\dot{y}$ ), and acceleration ( $\ddot{y}$ ),  $Z$  coordinate position ( $z$ ), velocity

**Table 2. Launch point prediction using UD filter & using smoother - simulated data**

( Point-A ( $x$ -true=1200.83,  $y$ -true=3001.06) at 0 s)

Case	Time at B (s)	Time at C (s)	U-D filter						Smoother					
			x-pre	x-std	% x err	y-pre	y-std	% y err	x-pre	x-std	% x err	y-pre	y-std	% y err
1	0	50.0	1200.4	1	0.03	2998	9.0	0.09	1199.9	0.2	0.08	2999.1	1.2	0.01
2	0.5	49.5	1200.3	2.6	0.04	2966.6	34.5	1.15	1199.9	0.2	0.01	2999.2	1.2	0.06
3	2.5	47.5	1238.4	16.2	-3.13	3186	103.4	-6.16	1200.4	0.3	0.04	3001.3	0.7	-0.01
4	5.0	45.0	1313.3	43.9	-9.36	3092	306	-3.03	1200.6	0.5	0.02	3000.2	1.6	0.03
5	10.0	40.0	1258.7	62.9	-4.82	4381.5	1049	-45.9	1199.8	0.8	0.08	3000.8	2.3	0.01
6	20.0	30.0	1357.1	104	-13	1569.7	1085	47.7	1200.6	1.4	0.02	2996.7	4.1	0.12

$x$ ,  $y$ -pre = predicted  $x$ ,  $y$ -position,  $x$ ,  $y$ -std = standard deviation of  $x$ ,  $y$ -position prediction,

$x$ ,  $y$ -err =  $x$ ,  $y$ -position error =  $((x, y\text{-true}) - (x, y\text{-pre}))$ , %  $x$ ,  $y$  err =  $(x, y\text{ err} * 100)/(x, y\text{-true})$ ,

$mx$ ,  $my$  = mean of  $x$ ,  $y$  residuals, PFE  $x$ ,  $y$  = Percentage fit error in  $x$ ,  $y$ -position.

**Table 3. Prediction of impact point using UD filter & launch point using smoother - real data**

Case	Time at B (s)	Time at C (s)	U-D filter						Smoother		
			x-std	PFE <sub>x</sub>	y-std	PFE <sub>y</sub>	z-std	PFE <sub>z</sub>	x-std	y-std	z-std
1	0	125	31.3	0.05	31.31	0.03	31.3	0.27	21.12	21.12	21.12
2	2.5	122.5	102.7	0.22	102.7	0.21	102.7	0.91	72.76	72.76	72.76
3	5	120	232.9	0.15	232.9	0.25	232.9	0.48	174.9	174.9	174.9
4	10	115	673.1	0.24	673.1	0.22	673.1	0.66	534.1	534.1	534.1

$x$ ,  $y$ -pre = predicted  $x$ ,  $y$ -position,  $x$ ,  $y$ -std = standard deviation of  $x$ ,  $y$ -position prediction,

$x$ ,  $y$ -err =  $x$ ,  $y$ -position error =  $((x, y\text{-true}) - (x, y\text{-pre}))$ , %  $x$ ,  $y$  err =  $(x, y\text{ err} * 100)/(x, y\text{-true})$ ,

$mx$ ,  $my$  = mean of  $x$ ,  $y$  residuals, PFE  $x$ ,  $y$  = Percentage fit error in  $x$ ,  $y$ -position.

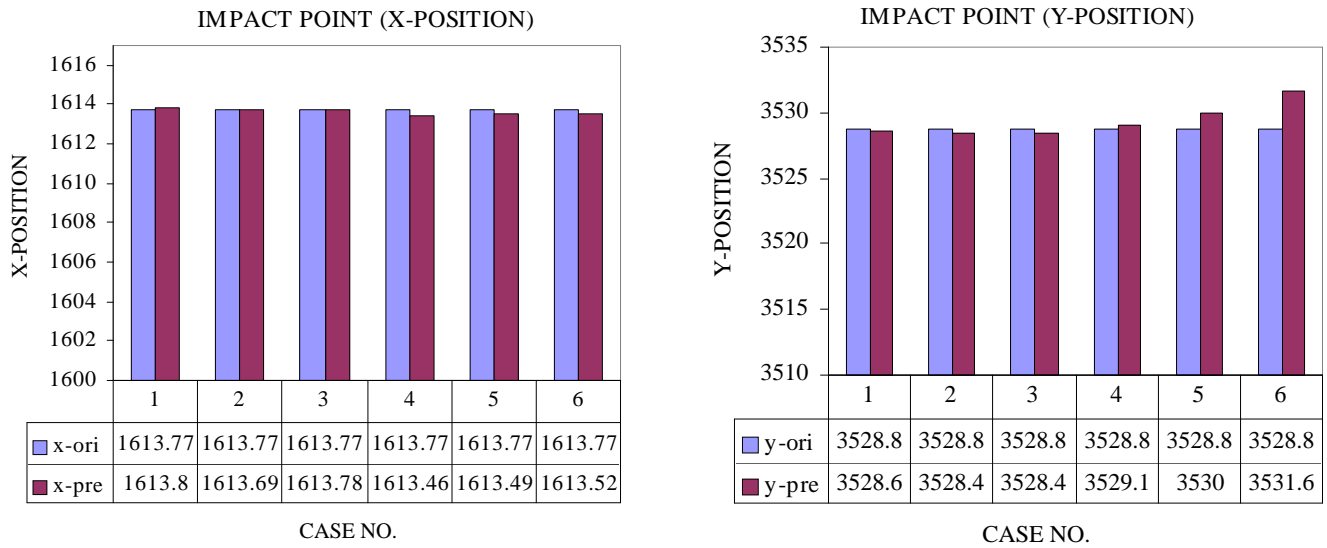


Figure 4(a). Impact point prediction using UD filter output.

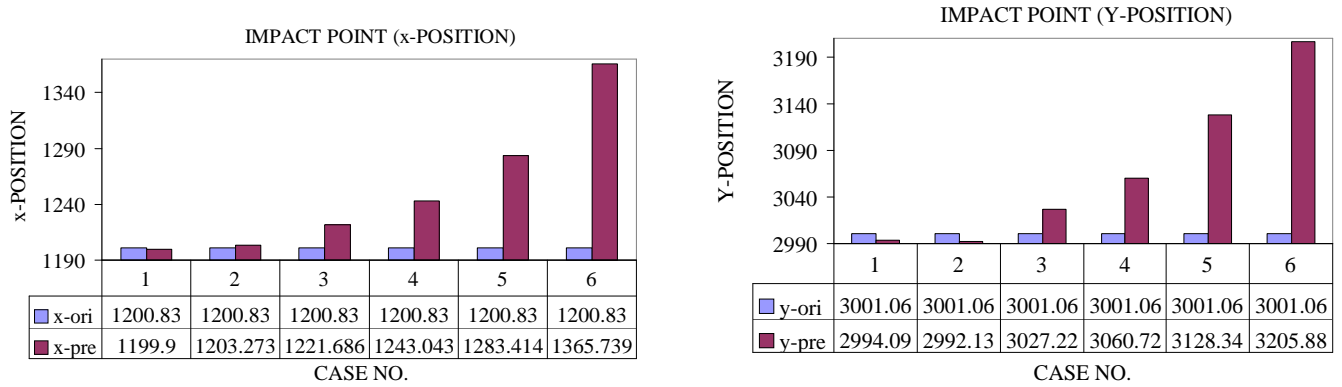


Figure 4(b). Launch point prediction using UD filter output.

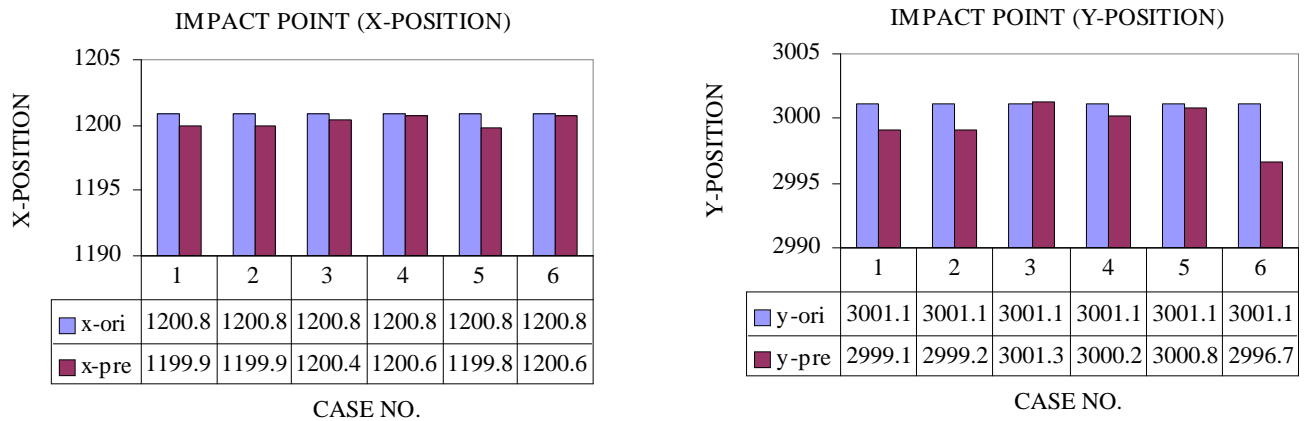


Figure 4(c). Launch point prediction using smoother.



( $\dot{z}$ ), and acceleration ( $\ddot{z}$ ) resulting in the 9 state vector  $[x, \dot{x}, \ddot{x}, y, \dot{y}, \ddot{y}, z, \dot{z}, \ddot{z}]$ . The model of the system is given by the Eqns (1) and (2). The transition matrix and the observation matrix are given by:

$$\Phi = \begin{bmatrix} 1 & \Delta T & \frac{\Delta T^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & \Delta T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta T & \frac{\Delta T^2}{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta T & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta T & \frac{\Delta T^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta T \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \quad (17)$$

The measurement noise covariance matrix  $R$  and process noise covariance matrix  $Q$  are estimated adaptively<sup>6</sup> using the following relations:

$$R = (\rho * \sigma_a)^2 \quad (18)$$

where  $\sigma_a$  is the angular accuracy of the radar and  $\rho$  is the estimated range and is given by

$$\rho = \text{sqrt}(\hat{x}_k^2 + \hat{y}_k^2 + \hat{z}_k^2), \quad k = N_B, \dots, N_C \quad (19)$$

and  $\hat{x}$ ,  $\hat{y}$  and  $\hat{z}$  are the estimated positions.

$$Q = (k_1 * \text{sqrt}(R) * e^{-k_2 * k * \Delta T})^2$$

$$k = N_B, \dots, N_C \quad (20)$$

where the proportionality factors  $k_1$  and  $k_2$  are chosen based on trial and error using post-flight measurement data.

Figure 5 shows the  $x$ ,  $y$  and  $z$ -position state estimations using the data between the points B and C (in left-half of the figure) assuming that the

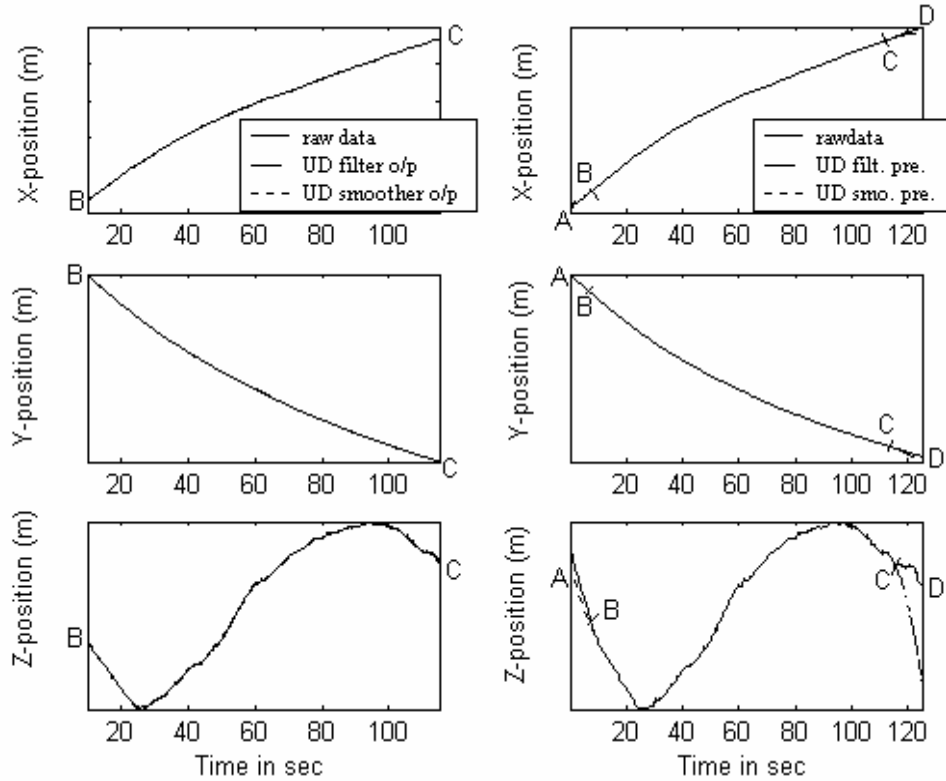


Figure 5.  $x$ ,  $y$  and  $z$ -position state estimations.

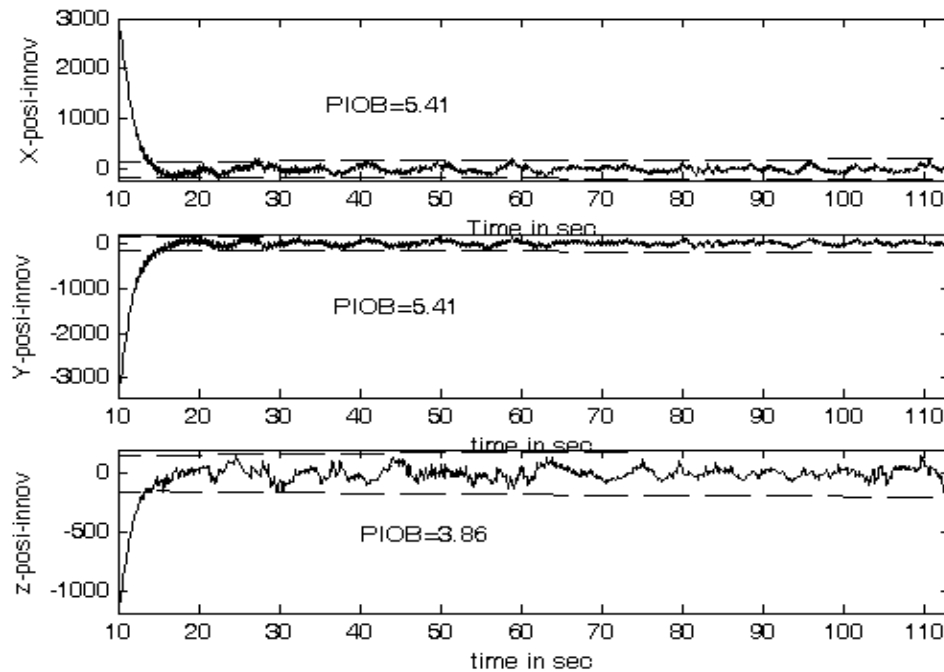


Figure 6. Innovation with bounds.

measured data is available between 10 s and 110 s. The state estimates with prediction of launch point (point A) using the smoother output and impact point (point D) using the filter output are compared with the true states in the right-half of the Fig. 5.

Table 3 gives the quantitative results of launch point and impact point estimation. Figs 6 and 7 show the innovations with  $2\sigma$  bounds and autocorrelation of residuals with bounds. Percentage innovation out of bounds (PIOB), percentage autocorrelation

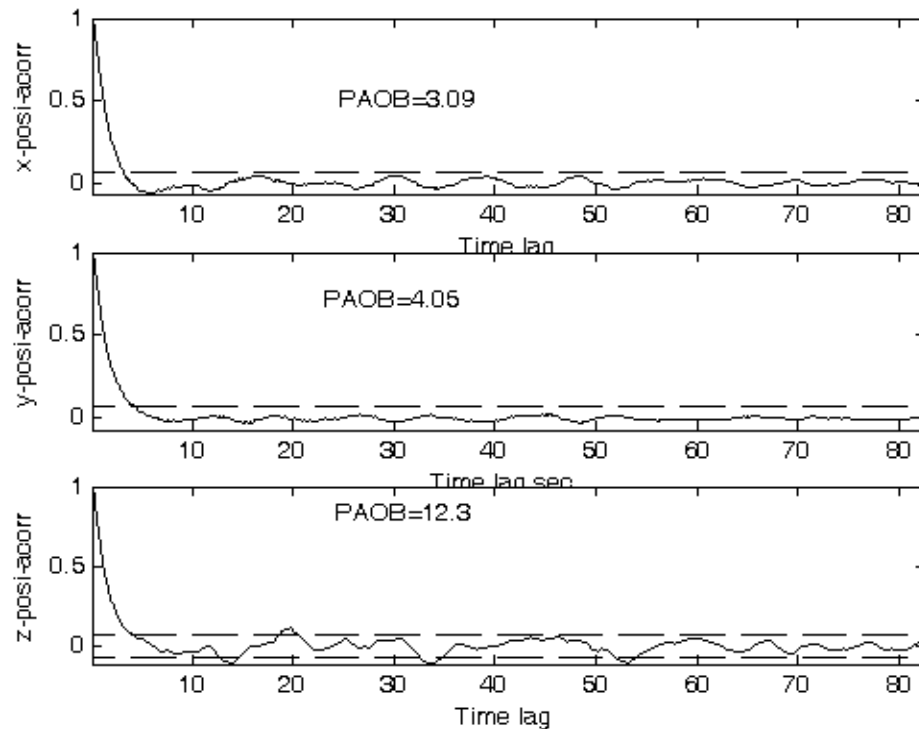


Figure 7. Autocorrelation with bounds.

out of bounds (PAOB) are close to the acceptable theoretical limit of 5 per cent, except for PAOB in  $z$ -position. This could be because of the manoeuvre observed in the  $z$ -position.

The autocorrelation function is well within the theoretical bounds. However, it also indicates that the measurement noise statistics may not be white,

and hence, inclusion of a model for this may improve the results. This is being investigated. The quantitative results of launch-point and impact-point estimations are shown in bar graph form in Figs 8(a) and 8(b). For this set of real data, it was observed that when the measurement data length observed by the sensors was  $< 115$  s, the launch-point estimation led to unacceptable results. The results indicate higher

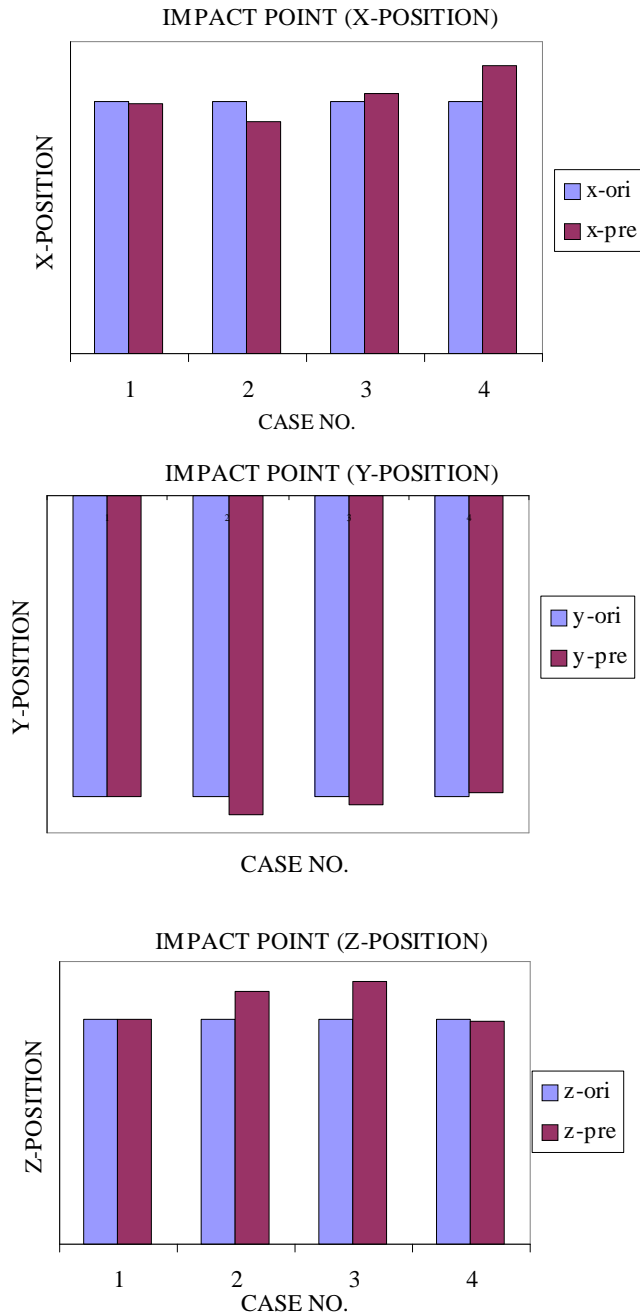


Figure 8(a). Impact point estimation for real data.

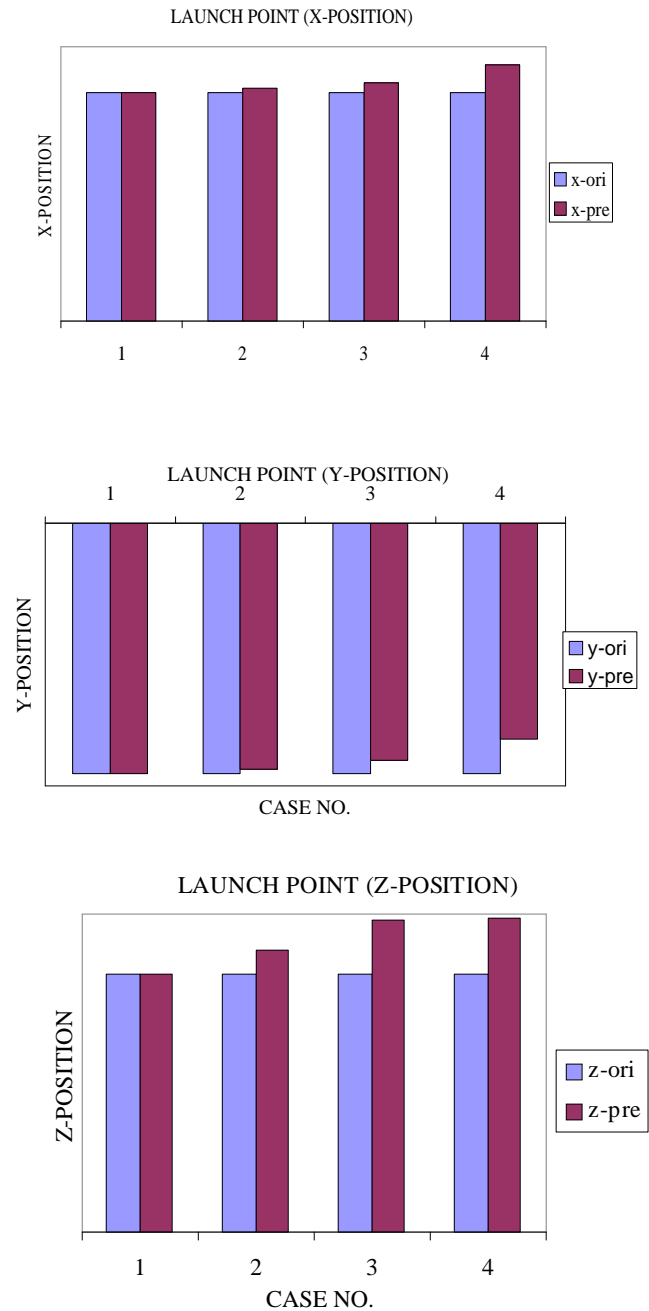


Figure 8(b). Launch point estimation for real data.

errors in impact-point prediction in the  $z$ -direction because of the manoeuvre in that axis, and if the measurements are missing during this portion, the prediction accuracy degrades considerably. Also, the launch point error is higher than the impact-point error.

The results for real data are for the case when the target is in the ballistic mode. In general when any target is launched, it is not in the ballistic mode and certain other factors like thrust and drag must be considered to get accurate estimates of both launch and impact points. Thus estimation of start point would require further study including these effects.

#### 4. CONCLUSIONS

The U-D factorised Kalman filter and R-T-S smoother were implemented in PC MATLAB and their performance studied using simulated data. The R-T-S smoother was found to generate accurate state estimates, which lead to better launch-point prediction accuracies. Impact point and launch-point prediction from real data of a guided target in ballistic mode was also evaluated. For real data, the inaccuracies in both launch-point and impact-point predictions can be reduced using improved models that account for drag and atmospheric effects. This requires further study. It will be worthwhile to explore the application of the presented approach to the cases where the state equations are highly nonlinear and the data spans are shorter.

#### ACKNOWLEDGEMENTS

The authors acknowledge with gratitude the technical discussions with Mr Appavu Raj, Scientist, Interim Test Range (ITR), Chandhipur.

#### 5. REFERENCES

1. Brookner, Eli. Tracking and Kalman filtering made easy. John Wiley and Sons, New York, 1998.
2. Gelb, Arthur. Applied optimal estimation, Ed. 7. The M.I.T Press, Massachusetts, 1982.
3. Candillo, G.P.; Mrstik, A.V. & Plambeck, Y. A track filter for reentry objects with uncertain drag. *IEEE Trans. Aerospace Elect. Syst.* April 1999. **35**(2). pp. 394-09.
4. Bierman G.J. A new computationally efficient fixed - interval, discrete-time smoother. *Automatica*. 1983, **19**(5), 505-11.
5. Candy, James V. Signal processing, The model-based approach. McGraw-Hill, Int. Edition, Singapore, 1987.
6. Raol, J.R. & Girija G., Evaluation of adaptive Kalman filtering methods for target tracking applications, *In AIAA Guidance, Navigation and Control Conference and Exhibit*, Montreal, Canada. Paper No. AIAA 2001-4106, 6-9 August 2001.

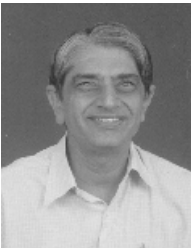
## Contributors



**Mr V.P.S. Naidu** obtained his ME (Medical Electronics) from the Anna University, Chennai, in 1997. He is working at National Aerospace Laboratories, Bangalore, as scientist since December 2001. His area of interest are: multi-sensor data fusion for target tracking, image registration, etc.



**Dr (Ms) Girija G** obtained her PhD from the Bangalore University in 1996. She is presently, working as a Scientist at the NAL, Bangalore. Her areas of research are: Modelling, parameter estimation of aerospace vehicles, and multi-sensor data fusion. She is a recipient of *NAL Foundation Day Award for Research* (1995). She has published about 50 research papers. She is a member of the Aeronautical Society of India.



**Dr J.R. Raol** obtained BE and ME both from the MS University, Baroda, in 1971 and 1974 and PhD from the McMaster University, Canada in 1986. He worked at the National Aerospace laboratories, Bangalore, from 1975 to 1981 and was actively involved in the Multidisciplinary Control Group's activities on human pilot modelling in fixed-and motion-based research simulators. He re-joined NAL in 1986 and is currently Head of the Flight Mechanics and Control Div of NAL. His current activities include modelling, identification, multi-sensor data fusion, fuzzy systems, genetic algorithm and neural networks. He is a senior member of the IEEE (USA), the fellow of IEE, life fellow of the Aeronautical Society of India, and life member of System Society of India. He has co-authored a book on Modelling and parameter estimation of dynamic system, published by IEE, UK, in 2004. He has 100 publications to his credit.